

**Vysoká škola báňská – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Mincovní jukebox pro Android**  
**Android Coin Jukebox**

**2014**

**Martin Stráňava**

## Zadání bakalářské práce

Student: **Martin Stráňava**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Mincovní jukebox pro Android**  
**Android Coin Jukebox**

### Zásady pro vypracování:

Cílem práce je vytvořit aplikaci typu jukebox pro mobilní platformu Android. Navržená aplikace bude schopna přehrávat videoklipy a zvukové záznamy uložené v knihovně, vytvářet seznamy oblíbených skladeb a třídit je podle různých kritérií. Toto řešení předpokládá připojení mincovníku pomocí Bluetooth, USB, nebo sériového rozhraní.

1. Rešerše obdobných aplikací pro Android.
2. Popis dostupných mincovníků a jejich API.
3. Připojení pomocí USB host, Bluetooth nebo sériového portu.
4. Implementace aplikace včetně komunikace s mincovníkem.
5. Testování aplikace na mobilních zařízeních (telefon a tablet).

### Seznam doporučené odborné literatury:

- [1] Reto Meier, Professional Android 2 Application Development, Wrox, 2010, ISBN-13: 978-0470565520  
[2] Sayed Hashimi, Pro Android 2, Apress, 2010, ISBN-13: 978-1430226598  
Cay S. Horstmann, Core Java(TM), Volume I--Fundamentals, Prentice Hall, 2007, ISBN-13: 978-0132354769

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Ing. Michal Krumník**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## Prehlásenie študenta

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave dňa: *4. mája 2014*

  
.....  
podpis študenta

## **Pod'akovanie**

Rád by som pod'akoval pánovi Mgr. Ing. Michalovi Krumniklovi za odbornú pomoc a konzultáciu pri vytváraní tejto bakalárskej práce, ďalej by som chcel pod'akovať rodičom a spolužiakom za morálnu podporu pri tvorbe tejto práce.

## **Abstrakt**

Cieľom tejto bakalárskej práce je zhotovenie zariadenia jukebox na platforme Android ako aj príslušnej aplikácie pre túto platformu. Zariadenie je schopné komunikovať s elektronickým mincovníkom, prehrávať audio a video záznamy. Oproti stávajúcim implementáciám jukebox-ov je výhodou hlavne cena za ktorú je možné tento jukebox zostaviť. Táto práca popisuje jednotlivé komponenty tvoriace dané zariadenie ako aj ich prepojenie. Funkčný celok je tvorený troma hlavnými časťami:

- FriendlyARM Smart210- vývojový kit pre platformu Android
- Mincovník- zariadenie pre akceptáciu mincí
- Arduino UNO- ako sprostredkovateľ informácie o vhodenej minci

Aplikácia s názvom Jukebox je napísaná v jazyku Java pomocou vývojového prostredia Eclipse, ktoré je oficiálne podporované spoločnosťou Google zastrešujúcou túto platformu.

## **Kľúčové slová**

Jukebox; Android; mincovník; Arduino UNO; FriendlyARM Smart210; Eclipse; Java; SOZA

## **Abstract**

The objective of this bachelor's thesis is create jukebox device with support Android platform. Device is capable to communicate with electronic coin acceptor and play the audio-video records. Beside existing implementations of this device, the main advantage is the price of whole system. This thesis describe components and their connections within the device. Functional unit consists of three main components:

- FriendlyARM Smart210- developer kit for Android platform
- Coin acceptor- device for coin acceptance
- Arduino UNO- mediator of information about inserted coin

Application itself is written in Java programming language, and developed in Eclipse, which is officially supported by Google company, owner of this platform.

## **Keywords**

Jukebox; Android; coin acceptor; Arduino UNO; FriendlyARM Smart210; Eclipse; Java; SOZA

## Zoznam použitých skratiek

Skratka	Význam
<b>3G</b>	Tretia generácia mobilných technológií
<b>ARM</b>	Acorn RISC Machine
<b>DC</b>	Direct Current
<b>GPS</b>	Global Positioning System
<b>SLC</b>	Single Level Cell
<b>NAND</b>	Not And
<b>TFT</b>	Thin Film Transistor
<b>LCD</b>	Liquid Crystal Display
<b>CPU</b>	Central Processor Unit
<b>RAM</b>	Random Access memory
<b>DDR</b>	Dual Data Rate
<b>EEPROM</b>	Electric Erasable Programable Read Only Memory
<b>I2C</b>	Inter Integrated Circuit
<b>SD</b>	Secure Digital
<b>UART</b>	Universal asynchronous receiver/transmitter
<b>USB</b>	Universal Serial Bus
<b>HDMI</b>	High-Definition Multimedia Interface
<b>FFT</b>	Flexible Flat Cable
<b>DVD</b>	Digital Versatile Disc
<b>API</b>	Application Programming Interface
<b>MS</b>	Microsoft
<b>IDE</b>	Integrated Development Environment
<b>PC</b>	Personal Computer
<b>SDRAM</b>	Synchronous dynamic random-access memory
<b>IO</b>	Input Output
<b>PWM</b>	Pulse-width modulation
<b>SPI</b>	Serial Peripheral Interface

---

<b>ICSP</b>	In Circuit Serial Programming
<b>NO</b>	Normal Open
<b>NC</b>	Normal Close
<b>TTL</b>	Transistor-Transistor Logic
<b>RS-232</b>	Štandard pre sériovú komunikáciu
<b>GPIO</b>	General-Purpose Input/Output
<b>SDK</b>	Software Development Kit
<b>TTY</b>	Teletype
<b>JNI</b>	Java Native Interface
<b>SOZA</b>	Slovenský Ochranný Zväz Autorský
<b>SR</b>	Slovenská Republika
<b>DPH</b>	Daň z pridanej hodnoty
<b>GUI</b>	Graphical User Interface
<b>URI</b>	Uniform Resource Identifier

---



# Obsah

1	Úvod.....	- 11 -
2	Popis jednotlivých hardvérových súčastí zariadenia.....	- 12 -
2.1	FriendlyARM Smart210.....	- 12 -
2.1.1	Špecifikácia Stamp modulu.....	- 12 -
2.1.2	Špecifikácia SDK-Board.....	- 13 -
2.1.3	Špecifikácia LCD displeja.....	- 14 -
2.1.4	Popis inštalácie systému Android 4.0.3 na zariadenie Smart210.....	- 14 -
2.2	Arduino UNO.....	- 16 -
2.2.1	Špecifikácia Arduino UNO.....	- 17 -
2.3	Elektronický mincovník.....	- 17 -
2.3.1	Rozdelenie mincovníkov.....	- 18 -
2.3.2	Zvolený mincovník COM-11719.....	- 19 -
2.3.3	Nastavenia mincovníka COM-11719.....	- 20 -
2.3.4	Postup pri programovaní mincovníka COM-11719.....	- 21 -
2.4	Prepojenie jednotlivých komponent zariadenia.....	- 22 -
2.4.1	Priebeh prenosu informácie z mincovníka do aplikácie Jukebox.....	- 22 -
2.4.2	Zdrojový kód programu nahraného v Arduino UNO.....	- 22 -
2.4.3	Alternatívne riešenia komunikácie mincovníka s aplikáciou Jukebox....	- 23 -
3	Aplikácia Jukebox z programátorského pohľadu.....	- 24 -
3.1	Popis aplikačnej logiky.....	- 24 -
3.1.1	JukeboxController.....	- 25 -
3.1.2	SongPlayer.....	- 25 -
3.1.3	SamplePlayer.....	- 25 -
3.1.4	AbstractCreditController.....	- 25 -
3.1.5	CreditController.....	- 26 -
3.1.6	MediaRetriever.....	- 26 -
3.1.7	TopSongDataSource.....	- 26 -
3.1.8	Artist.....	- 27 -

3.1.9	Album.....	- 27 -
3.1.10	MediaEntity .....	- 27 -
3.2	Prepojenie aplikačnej logiky s aplikáciou Jukebox.....	- 27 -
3.2.1	MainActivity .....	- 28 -
3.2.2	AdminActivity.....	- 28 -
3.2.3	Application .....	- 28 -
3.3	Android SerialPort API .....	- 28 -
3.3.1	Podporované funkcie API.....	- 28 -
3.3.2	Nepodporované funkcie API .....	- 28 -
3.4	Vynútené spustenie a beh aplikácie Jukebox .....	- 29 -
3.4.1	Funkčnosti aplikácie SureLock využité v Jukeboxe.....	- 29 -
4	Aplikácia Jukebox z užívateľského pohľadu .....	- 30 -
4.1	Užívateľské rozhranie aplikácie Jukebox.....	- 30 -
4.2	Administrátorské rozhranie aplikácie Jukebox .....	- 31 -
4.2.1	Záložka SECURITY.....	- 31 -
4.2.2	Záložka COMMUNICATION.....	- 32 -
4.2.3	Záložka COIN ACCEPTOR.....	- 32 -
4.2.4	Záložka JUKEBOX.....	- 33 -
5	SOZA- Slovenský ochranný zväz autorský.....	- 34 -
5.1	Sadzobník pre hudobné automaty platný v roku 2014 .....	- 34 -
5.2	Špeciálne označenie hudobných automatov .....	- 35 -
	Záver .....	- 36 -
	Použitá literatúra .....	- 37 -
	Zoznam príloh .....	- 38 -

# 1 Úvod

Cieľom mojej bakalárskej práce bolo zhotoviť zariadenie jukebox, komunikujúce s elektronickým mincovníkom na platforme Android. Táto platforma bola zvolená z dôvodu širokej škály možností, čo sa týka podporovaných zariadení. Vďaka tomu je možné takéto zariadenie zostaviť za relatívne nízku cenu.

V druhej kapitole práce je podrobne rozpísaná hardvérová špecifikácia jednotlivých komponentov a to FriendlyARM Smart210, Arduino UNO a elektronického mincovníka COM-11719. Pri zariadení Smart210 je popísaný postup pre inštaláciu operačného systému Android 4.0.3. Pri komponente Arduino UNO okrem hardvérovej špecifikácie je uvedený zdrojový kód programu, ktorý sprostredkováva informáciu o vhodenej minci. Ohľadom elektronického mincovníka je popísané základné rozdelenie, nastavenie a programovanie tohto zariadenia. Záver kapitoly sa zaoberá vzájomným prepojením komponentov a interakcií medzi nimi. Taktiež predkladá alternatívne možnosti komunikácií jukeboxu s mincovníkom.

Tretia kapitola opisuje aplikáciu Jukebox z programátorského hľadiska. V úvode tejto kapitoly je podrobne opísaná samotná aplikačná logika aplikácie Jukebox a to podrobným opisom tried a ich vzájomných prepojení. Ďalej táto kapitola opisuje prepojenie aplikačnej logiky s jednotlivými aktivitami aplikácie Jukebox. Keďže Android SDK neobsahuje API pre komunikáciu prostredníctvom sériového portu, bolo potrebné dodať toto aplikačné rozhranie do aplikácie Jukebox a to konkrétne Android SerialPort API. Toto aplikačné rozhranie je popísané v predposlednej časti kapitoly. V poslednej časti kapitoly je vysvetlené vynútené spustenie aplikácie Jukebox prostredníctvom aplikácie SureLock.

Štvrtá kapitola práce poukazuje na aplikáciu Jukebox z pohľadu užívateľa. Užívateľské rozhranie aplikácie je rozdelené na dve základné časti. Prvá časť je užívateľské rozhranie pre ovládanie prehrávania a druhá časť - administrátorské rozhranie, je určené pre nastavovanie parametrov aplikácie Jukebox.

Piata kapitola je venovaná Slovenskému ochrannému zväzu autorskému (SOZA). Keďže pre prevádzku jukebox-ov ako takých je potrebné vybaviť licenciu, táto kapitola sa zaoberá danou problematikou. Konkrétne pre Slovenskú republiku. Rieši aké pravidlá sú pre prideľovanie licenčných znáмок na zariadenia. V tejto kapitole sú vypísané orientačné ceny mesačných poplatkov.

---

## 2 Popis jednotlivých hardvérových súčastí zariadenia

V tejto kapitole je podrobne popísaná hardvérová špecifikácia jednotlivých zariadení, prípadne popis nastavovania alebo inštalácia softvéru.

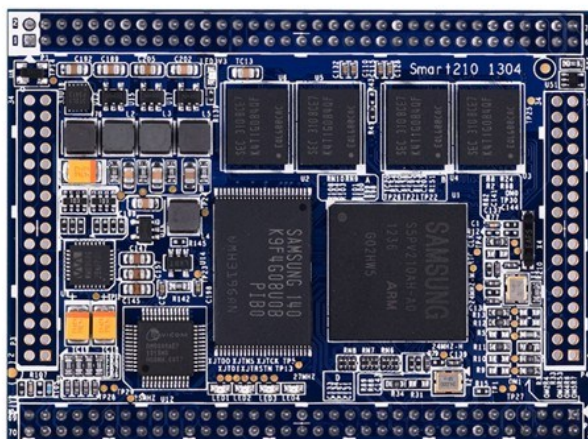
### 2.1 FriendlyARM Smart210

FriendlyARM Smart210 [1] (ďalej len Smart210) je vývojový kit, postavený na architektúre ARM. Vývojári kládli dôraz na nízku cenu pri relatívne vysokom výkone. Zariadenie môže byť vďaka bohatej výbave periférií použité ako napríklad internetový terminál, multimediálny terminál, súčasť inteligentnej domácnosti, súčasť monitorovacieho systému a iné. Podporované operačné systémy sú: Android 4.0.3, Android 3.2, Debian 6 a Windows CE 6. Toto zariadenie som zvolil kvôli jeho cenovej dostupnosti, bohatej výbave periférií a možnosti pridávania rôznych modulov (Bluetooth modul, WiFi modul, 3G modul, GPS modul...) Smart210 ako celok je tvorený troma časťami:

- Smart210 SDK-Board na ktorej sú umiestnené všetky periférie (obrázok 2.2)
- Stamp Module obsahujúci procesor, operačnú pamäť a SLC NAND Flash pamäť (obrázok 2.1)
- Multi-dotikový kapacitný TFT LCD displej (obrázok 2.3)

#### 2.1.1 Špecifikácia Stamp modulu

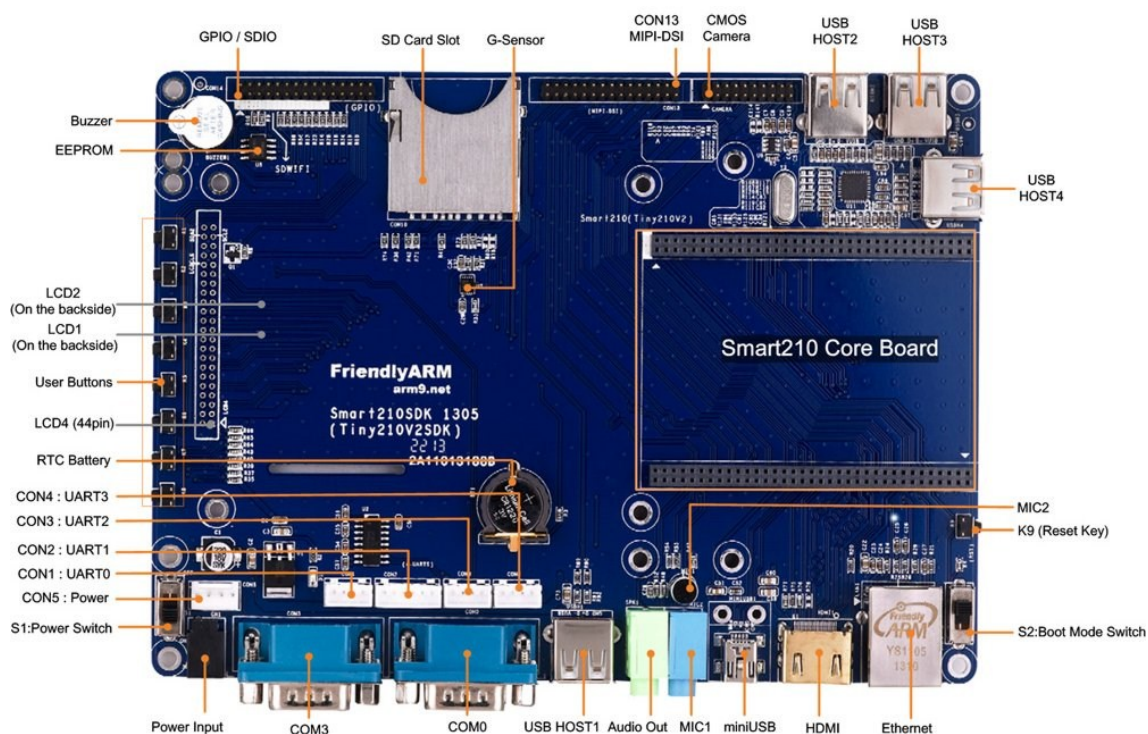
- CPU: Samsung S5PV210 Cortex A8 1GHz s grafickým akceleratorom PowerVR SGX540
- RAM: DDR2 512MB, 32bit-ová šírka zbernice, na takte 200MHz
- Flash: 512MB SLC NAND Flash



Obrázok 2.1: Stamp modul

### 2.1.2 Špecifikácia SDK-Board

- EEPROM: 256 Byte prístupnú prostredníctvom rozhrania I2C
- Externá pamäť: soket pre SD kartu
- Sériové porty: dva krát konektory DB9 (RS232), štyri krát UART umiestnené na doske a štyri krát USB host
- Audio výstup: 3.5mm jack
- Audio vstup: 3.5mm jack
- Sieťové rozhranie: RJ-45 10/100M
- Video výstup: HDMI a dva FFC konektory určené pre pripojenie LCD displejov
- Užívateľské vstupy:
- Napájanie: 5V



Obrázok 2.2: Smart210 SDK-Board

---

### 2.1.3 Špecifikácia LCD displeja

- Technológia: TFT LCD
- Uhlopriečka: 7"
- Rozlíšenie: 800x480
- Podporovaných súčasných dotykov: 5
- Technológia dotyku: kapacitný



Obrázok 2.3: LCD displej namontovaný na SDK-Board

### 2.1.4 Popis inštalácie systému Android 4.0.3 na zariadenie Smart210

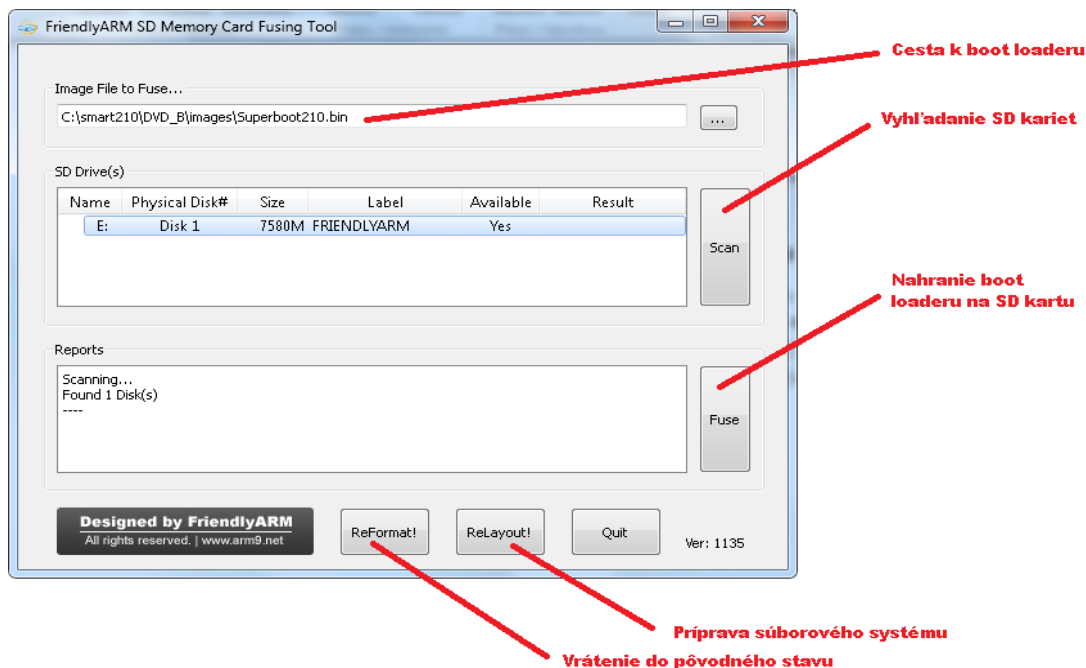
Na zariadení Smart210 nebol predinštalovaný operačný systém, preto ho bolo nutné nainštalovať. Obsahom balenia bolo aj DVD s obrazmi podporovaných systémov. Pri výbere systému som mal 2 možnosti: svižnejší Android 3.2 ktorý má API level 13 a Android 4.0.3 s API level 15. Pre aplikáciu Jukebox som zvolil operačný systém Android 4.0.3.

#### 2.1.4.1 Príprava karty SD pred inštaláciou

Na prípravu karty SD slúži utilita SD-Flasher ktorá je dostupná na DVD pribalenom k zariadení. SD-Flasher je podporovaný iba pre platformu MS Windows. Postup pri príprave SD karty je nasledovný:

- Kartu SD vložíme do počítača
- Spustíme program SD-Flasher.exe (obrázok 2.4) s administrátorskými právami
- Stlačíme tlačidlo 'Scan' pre vyhľadanie dostupných pamäťových kariet
- V sekcii 'SD Driver(s)' označíme žiadané médium
- Stlačíme tlačidlo 'ReLayout!' - tým sa pripraví súborový systém na médiu

- V sekcii 'Image File to Fuse...' nadefinujeme cestu k súboru Superboot210.bin, ktorý je umiestnený na pribalenom DVD
- Stlačíme tlačidlo 'Fuse' pre nahranie boot loaderu na médium
- Pre obnovenie súborového systému na danom médiu do pôvodného stavu slúži tlačidlo 'ReFormat!'. Túto funkciu môžeme využiť po úspešnej inštalácii
- Nakoniec program ukončíme tlačidlom 'Quit'



Obrázok 2.4: Popis aplikácie SD-Flasher

#### 2.1.4.2 Nahranie obrazu systému Android 4.0.3 na kartu SD

Nahranie obrazu systému Android na SD kartu je možné previesť v ľubovoľnom súborovom manažéri napríklad Windows Explorer. Postupujeme nasledovne:

- Na karte SD vytvoríme v koreňovom adresári adresár /images
- Do adresára /images na karte SD skopírujeme priečinok Android obsahujúci obraz operačného systému, ktorý je umiestnený na pribalenom DVD v adresári /images
- Ďalej do adresára /images skopírujeme konfiguračný súbor FriendlyARM.ini, ktorý je umiestnený na pribalenom DVD v adresári /images
- Kartu SD bezpečne odpojíme

#### 2.1.4.3 Samotná inštalácia systému Android 4.0.3

Zariadenie Smart210 podporuje dva módy zavádzania systému. Prvý 'SD Card Mode', ktorý umožňuje zavádzanie systému z karty SD a druhý 'NAND Flash Mode', ktorý umožňuje zavádzanie systému priamo z integrovanej NAND Flash pamäte. Tieto módy je možné prepínať prepínačom S2 umiestnenom na SDK-Board. V postupe pre inštaláciu predpokladáme, že zariadenie je pripojené na zdroj napájania. Postup inštalácie:

- 
- Vopred pripravenú kartu SD s obrazom systému vložíme do zariadenia Smart210
  - Prepínač S2 prepneme do polohy 'SD Card Mode'
  - Zariadenie zapneme prepínačom S1
  - Zavádzač systému automaticky skopíruje a dekomprimuje obraz systému Android do pamäte NAND Flash
  - Po dokončení operácie prepneme S2 do 'NAND Flash Mode' a zariadenie vypneme prepínačom S1
  - Pri ďalšom spustení zariadenia sa zavedie Android 4.0.3

## 2.2 Arduino UNO

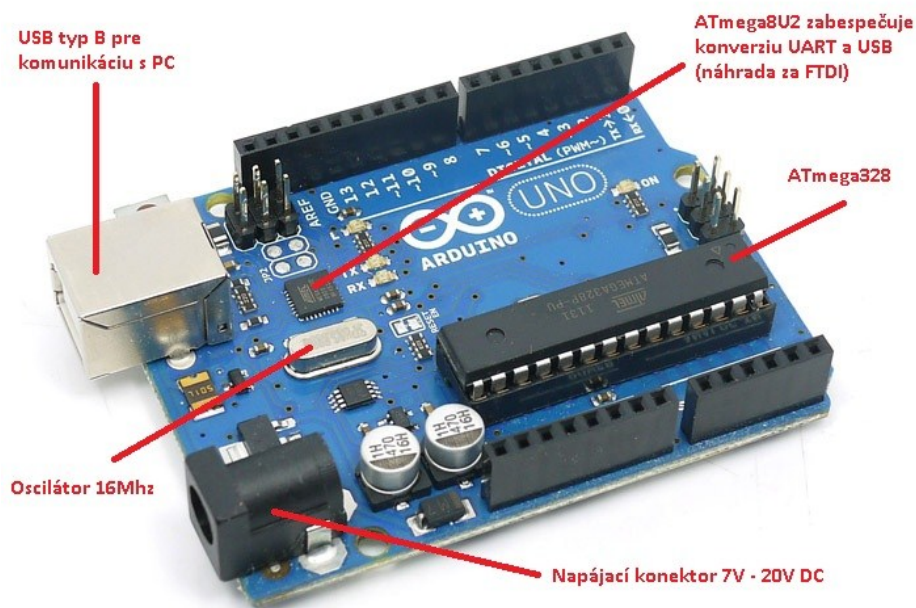
Arduino Uno [2] je open-source platforma založená na 8-bitovom mikrokontroléri ATmega328 od firmy Atmel a grafickom vývojovom prostredí Arduino IDE. Arduino môže byť použité k vytváraniu samostatných interaktívnych zapojení alebo môže byť pripojené k softvéru na počítači (napr. Macromedia Flash). Zariadenie je znázornené na obrázku 2.5.

Projekt vznikol v roku 2005 v Taliansku v meste Ivrea. Jeho cieľom bolo vytvoriť jednoduchú prototypovaciu platformu pre študentov, ktorá umožní rýchly vývoj a jednoduché používanie. Za jeho vývojom a predají stojí firma Smart Projects. Projekt zaznamenal veľký úspech a neskôr začali vznikať novšie verzie. Do februára 2010 sa údajne predalo viac ako 120 tisíc kusov.

Projekt Arduino je od svojho počiatku voľne dostupný (open-source) všetkým užívateľom, ktorí ho chcú používať alebo vylepšovať. Dokumentácia a referenčná príručka jazyka a externých knižníc je väčšinou vydaná pod licenciou Creative Commons. Vďaka tomu sú k dispozícii všetky zdrojové kódy a je dodržiavaná určitá kompatibilita. Arduino sa snaží byť taktiež cenovo dostupné čo je dosiahnuté pomocou lacnej pracovnej sily v Taliansku. Pre svoje univerzálne uplatnenie je Arduino vyhľadávané predovšetkým domácimi kutilmi. Možnosti jeho využitia sú rozsiahle.

Komunikácia s počítačom je riešená prostredníctvom rozhrania USB typ B, kde konverziu medzi technológiami UART a USB zabezpečuje mikrokontrolér ATmega8U2 s vopred nameraným firmvérom. Zdrojové kódy firmvéru sú voľne dostupné. V starších verziách túto komunikáciu zabezpečoval obvod FTDI. Aj keď je komunikácia s PC vedená cez rozhranie USB, je softvérovo simulovaná sériová komunikácia cez linku RS-232. Ovládače pre komunikáciu s PC sú naimplementované pre väčšinu platforiem.





Obrázok 2.5: Arduino UNO s popisom niektorých častí

### 2.2.1 Špecifikácia Arduino UNO

- Mikrokontrolér: ATmega328
  - Taktovacia frekvencia: 16Mhz (externý keramický oscilátor)
  - Veľkosť SDRAM: 2kB
  - EEPROM: 1kB
  - Flash: 32kB
  - Digitálne I/O: 14 z toho 6 môže byť PWM výstup
  - Analógové vstupy: 6
- Komunikačné rozhrania: USB, SPI, I2C, UART, ICSP
- Napájacie napätie 7V - 20V DC

## 2.3 Elektronický mincovník

Elektronický mincovník je zariadenie, ktoré vyhodnocuje fyzikálne parametre mince alebo žetónu. Tieto fyzikálne parametre môžu byť napríklad: rozmer, hmotnosť alebo elektromagnetické vlastnosti. Podľa týchto parametrov rozhodne, či vhozená minca alebo žetón vyhovuje alebo nevyhovuje požadovaným parametrom. Podľa toho ju buď akceptuje alebo nie. Na trhu je nespočetné množstvo takýchto zariadení a líšia sa hlavne spôsobom prenosu informácie o akceptovanej minci alebo žetóne. Ďalším parametrom je akým spôsobom zariadenie mince akceptuje. Mincovníky sa líšia aj spôsobom programovania (učenia).

---

### 2.3.1 Rozdelenie mincovníkov

Na trhu je množstvo mincovníkov, ktoré majú rôzne parametre a vlastnosti. V nasledujúcom texte je popísané základné rozdelenie týchto zariadení:

#### 2.3.1.1 Rozdelenie mincovníkov podľa spôsobu porovnávania parametrov

- Porovnávanie prostredníctvom pevného prototypu vloženého do mincovníku. Takéto mincovníky sú síce lacnejšie ale ich hlavnou nevýhodou je, že dokážu akceptovať iba jeden druh mince alebo žetónu.
- Porovnávanie prostredníctvom uložených informácií, ktoré boli zhromaždené (naučené) pri programovaní zariadenia. Výhodou tohto typu je, že takýto mincovník zväčša umožňuje zapamätanie viacerých druhov mincí. Rádovo je to 3, 6, 8 alebo 12 mincí. Takéto mincovníky môžu podporovať aj prepínanie medzi viacerými menami. Toto prepínanie je väčšinou možné pomocou hardvérového prepínača. Napríklad mincovník EU2 umožňuje prepínanie medzi dvoma menami s tým, že v každej mene si dokáže pamätať šesť mincí. Toto riešenie je vhodné hlavne v pohraničných oblastiach.

#### 2.3.1.2 Rozdelenie mincovníkov podľa spôsobu komunikácie

- Pulzný výstup- Zariadenie vygeneruje sekvenciu pulzov. Počet týchto pulzov je buď nadefinovaný výrobcom alebo je umožnené užívateľovi tento počet nastaviť pri programovaní. Zariadenie umožňuje prepínanie medzi dvoma módmí a to NO alebo NC. Tento mód určuje aká logická úroveň je na výstupe v pokojovom stave. V prípade NO (Normal Open) je to logická 1 pričom v prípade NC (Normal Close) je to logická 0. Zariadenie ďalej umožňuje nastavenie dĺžky. Nastavenia týchto parametrov sú väčšinou umožnené hardvérovými prepínačmi umiestnenými na mincovníku.
- ccTalk [3]- je sériový protokol používaný hlavne v oblasti peňažných transakcií. Protokol bol vyvinutý spoločnosťou Coin Control, ktorá v súčasnosti existuje pod názvom Money Control. Prvá verzia protokolu vznikla v roku 1996. Technológia využíva asynchrónny prenos znakov podobne ako RS232 s tým rozdielom, že využíva práve jeden dátový vodič pre obojsmernú half-duplex komunikáciu. Napäťová úroveň je kompatibilná s TTL. Na jednej zbernici môže byť pripojených viac zariadení, pričom každé zariadenie má svoju vlastnú adresu. Každý paket obsahuje zdrojovú aj cieľovú adresu. Podľa cieľovej adresy sa rozoznáva, komu daný paket patrí. Zariadenie s inou adresou paket zahodí.
- RS232- sériová asynchrónna komunikácia. Protokol prenosu je závislý na konkrétnom mincovníku. Mincovník má väčšinou pevne stanovený nemenný baud rate (napr. 9600)

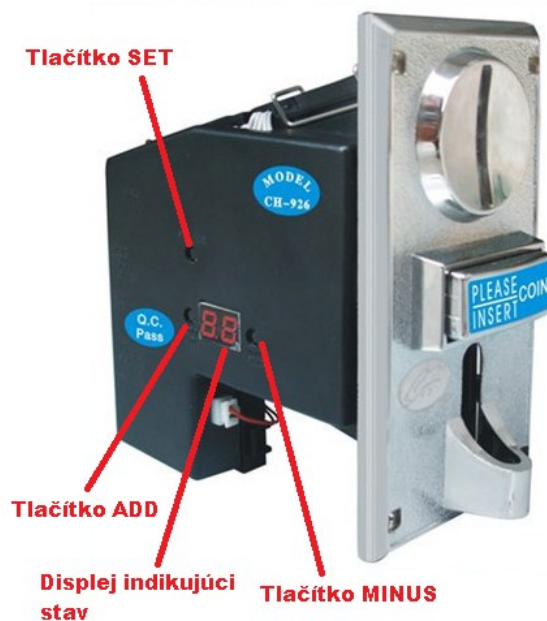
### 2.3.1.3 Rozdelenie mincovníkov podľa spôsobu programovania

- Programovanie prostredníctvom PC- k takémuto druhu mincovníku je dodaný softvér, pomocou ktorého sa mincovník programuje.
- Programovanie bez nutnosti spojenia s PC- takýto mincovník má špeciálne hardvérové tlačidlo alebo prepínač, pomocou ktorého sa prepína do režimu "učenia". Pribeh programovania takéhoto mincovníku je závislý na konkrétnom type.

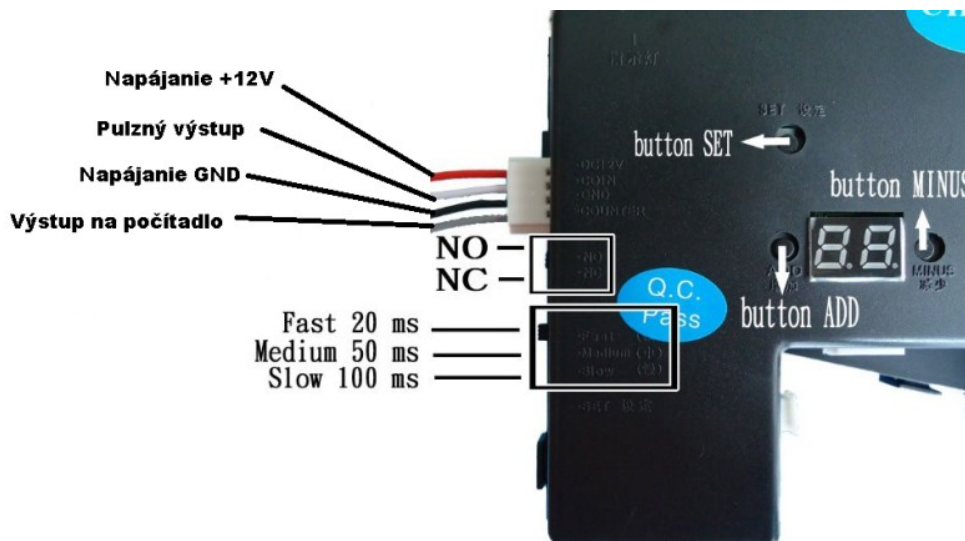
### 2.3.2 Zvolený mincovník COM-11719

COM-11719 zobrazený na obrázku 2.6 má nasledujúce vlastnosti:

- Má pamäť na tri mince
- Výstup je pulzný s programovateľným počtom pulzov na každú naučenú mincu
- Programuje sa bez použitia PC
- Napájacie napätie: 12V DC
- Možnosť prepínania NO/NC: Áno
- Možnosť nastavenia dĺžky pulzu: Áno



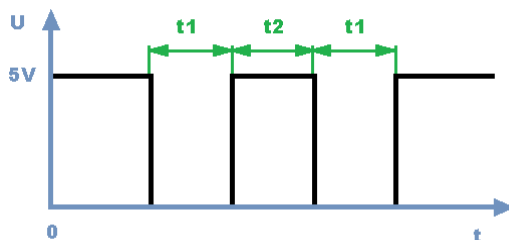
Obrázok 2.6: Mincovník COM-11719



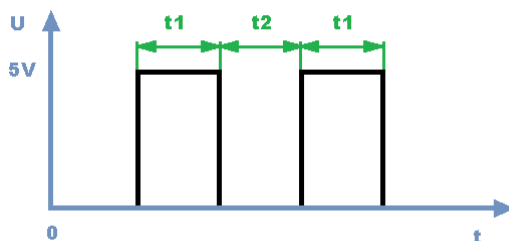
Obrázok 2.7: Popis prepínačov a konektora mincovníka COM-11719

### 2.3.3 Nastavenia mincovníka COM-11719

Na zariadení sú umiestnené dva prepínače znázornené na obrázku 2.7. Prvý prepínač NO/NC umožňuje nastavenie úrovne na výstupe v pokojovom stave. Možnosť NO určuje pokojový stav na úroveň logická 1 (obrázok 2.8), pričom NC určuje pokojový stav na úroveň logická 0 (obrázok 2.9). Druhý prepínač umožňuje zadefinovať dĺžku pulzu v milisekundách a to 20, 50 alebo 100 milisekúnd. V nasledujúcich dvoch obrázkoch bude táto hodnota označená ako  $t_1$ . Hodnota  $t_2$  je nemenná a je daná výrobcom.



Obrázok 2.9: Priebeh dvoch pulzov NO



Obrázok 2.8: Priebeh dvoch pulzov pri NC

---

## 2.3.4 Postup pri programovaní mincovníka COM-11719

### 2.3.4.1 *Začiatok programovania*

- Pripojíme mincovník na napájanie
- Stlačíme spínač ADD a MINUS naraz
- Na displeji sa zobrazí znak 'A'
- Stlačíme tlačidlo SET
- Na displeji sa zobrazí znak 'E'
- Tlačidlami ADD a MINUS si nastavíme, počet druhov mincí ktoré chceme uložiť (v našom prípade 3)
- Stlačíme tlačidlo SET

### 2.3.4.2 *Nastavenie hodnôt pre jednotlivé mince*

V texte sa vyskytuje parameter X ktorý je daný poradím mincí. Poradie je nasledovné: 1 pre 0,10€; 2 pre 0,20€ a 3 pre 0,50€. Táto časť postupu sa pre každú mincu opakuje:

- Na displeji sa zobrazí 'HX'
- Tlačidlami ADD a MINUS si nastavíme, koľko krát chceme vhodit' ukázkový druh mince (odporúčaná minimálna hodnota je 15)
- Stlačíme tlačidlo SET
- Na displeji sa zobrazí 'PX'
- Tlačidlami ADD a MINUS si nastavíme, koľko pulzov má zariadenie vygenerovať pre túto mincu (1)
- Stlačíme tlačidlo SET
- Na displeji sa zobrazí 'FX'
- Tlačidlami ADD a MINUS si nastavíme presnosť s akou má zariadenie danú mincu akceptovať (10)
- Stlačíme tlačidlo SET

### 2.3.4.3 *Dokončenie nastavení pre jednotlivé mince*

- Na displeji sa zobrazí 'A'
- Stlačíme tlačidlo SET
- Na displeji sa zobrazí 'E' čo značí, že nastavenie parametrov je dokončené
- Mincovník odpojíme od napájania
- Mincovník pripojíme k napájaniu

### 2.3.4.4 *Vhadzovanie jednotlivých vzoriek*

V tejto časti postupu je taktiež použitý hore popísaný parameter X a postup takisto opakujeme celkom tri krát:

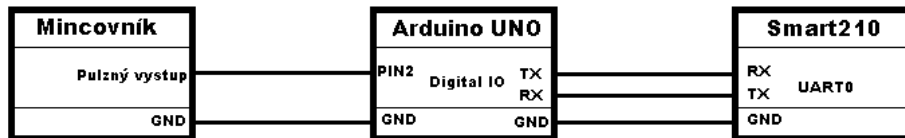
- Stlačíme tlačidlo SET
- Na displeji sa zobrazí 'AX'
- Následne do mincovníka vhadzujeme opakovane X-tý druh mince (0,10€). Počet opakovaní sme si vopred nadefinovali (HX)
- Po prevedení týchto opakovaných vhadzovaní na displeji bliká hodnota 'AX'
- Stlačíme tlačidlo SET (po prevedení poslednej iterácie je programovanie dokončené)

---

## 2.4 Prepojenie jednotlivých komponent zariadenia

- Pulzný výstup mincovníka je pripojený na PIN2 Arduino UNO
- Arduino UNO a Smart210 sú prepojené sériovou linkou prostredníctvom UART
- Všetky tri zariadenia majú spoločnú zem

Toto prepojenie je zobrazené na obrázku 2.10.



Obrázok 2.10: Bloková schéma prepojenia jednotlivých komponent

### 2.4.1 Priebeh prenosu informácie z mincovníka do aplikácie Jukebox

Mincovník pri vhození mince 0,10€; 0,20€ alebo 0,50€ vygeneruje 1, 2 alebo 5 pulzov. V zariadení Arduino UNO je nahraný program, ktorý reaguje pri externom prerušení na PIN2 a to konkrétne na nábežnú hranu. Obslužná rutina tohto prerušenia pošle cez sériovú linku reťazec "100X" na zariadenie Smart210. Aplikácia Jukebox, bežiaca na Smart210 reaguje na tento reťazec ako na podnet pre zvýšenie kreditu.

### 2.4.2 Zdrojový kód programu nahraného v Arduino UNO

Ako prvá sa vykoná funkcia setup(), ktorá je zodpovedná za inicializáciu programu. V tejto funkcii sa zahájí komunikácia prostredníctvom sériového portu s baud rate 9600. Následne sa pripojí obslužná rutina blink() na prerušenie 0 (PIN2). Funkcia blink() priradzuje hodnotu HIGH premennej state. Funkcia loop() sa vykonáva v nekonečnom cykle. Ak platí podmienka state je rovné HIGH program odošle cez sériový port reťazec "100X" a nastaví premennú state na hodnotu LOW. Popísané funkcie sú zobrazené ako zdrojový kód na obrázku 2.11.

---

```
volatile int state = LOW;

void setup() {
    Serial.begin(9600);
    attachInterrupt(0, blink, RISING); //interrupt 0: PIN2
}
void loop() {
    if(state == HIGH) {
        Serial.print("100X");
        state = LOW;
    }
}
void blink() {
    state = HIGH;
}
```

---

Obrázok 2.11: Zdrojový kód programu nahraného v Arduino UNO

---

### 2.4.3 Alternatívne riešenia komunikácie mincovníka s aplikáciou Jukebox

Jedným z alternatívnych riešení by mohlo byť pripojenie mincovníka priamo na GPIO zariadenia Smart210. Ďalším riešením je možnosť použiť mincovník ktorý disponuje sériovým výstupom a pripojiť ho priamo na sériový port Smart210. V prípade použitia zariadenia pre beh aplikácie Jukebox, ktoré nedisponuje sériovým rozhraním:

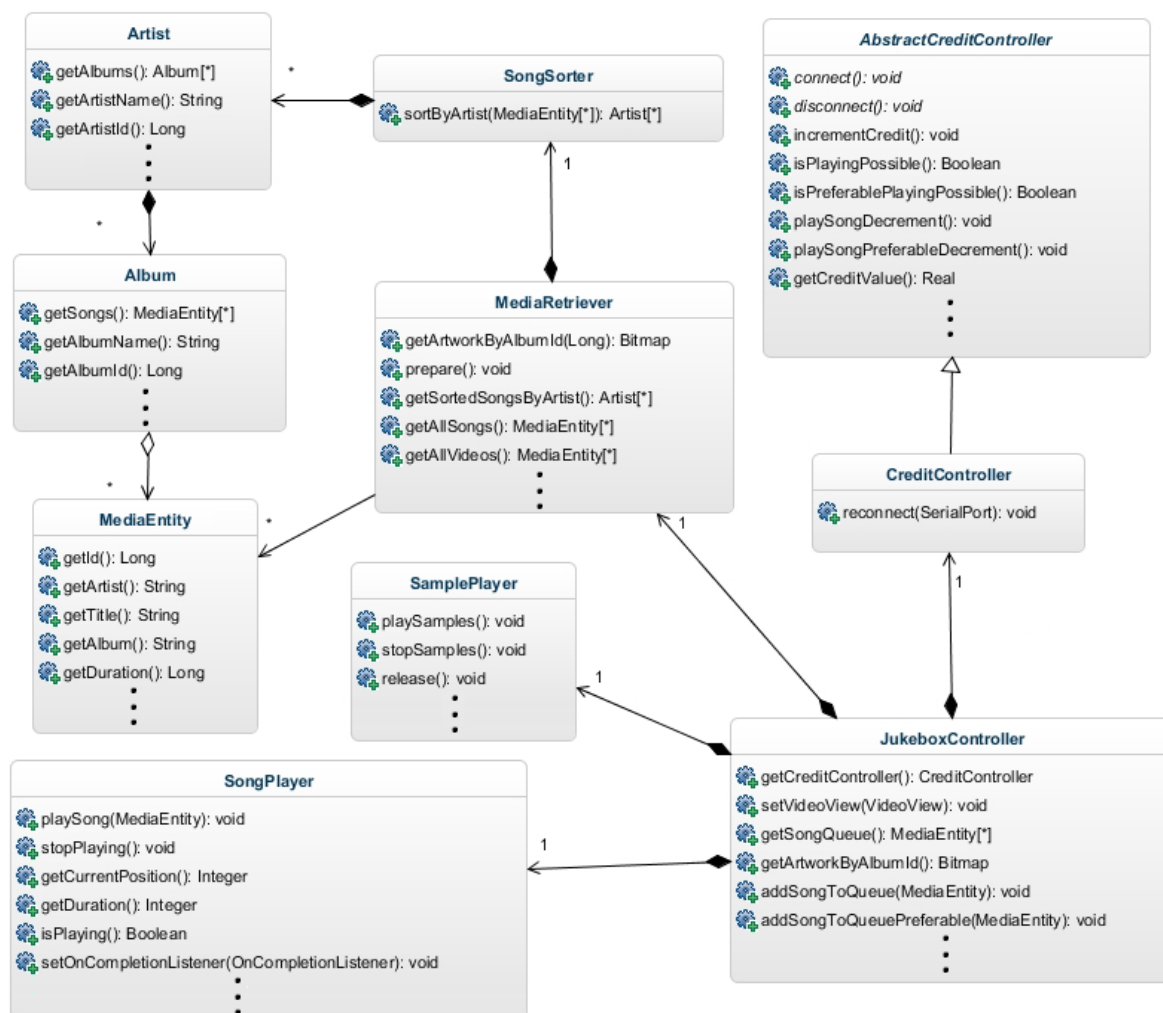
- Pridať Bluetooth modul k Arduino UNO a komunikovať prostredníctvom tejto bezdrôtovej technológie
- Pripojiť Arduino UNO prostredníctvom USB host

### 3 Aplikácia Jukebox z programátorského pohľadu

Aplikácia Jukebox je napísaná pre platformu Android s minimálnym API level 13. Vyžaduje super-užívateľské práva z dôvodu použitia Android SerialPort API. Aplikácia je tvorená dvoma aktivitami a to MainActivity a AdminActivity. Celá aplikačná logika je zabalená v jednom objekte JukeboxController. Nasledujúce podkapitoly vysvetľujú funkcie jednotlivých objektov a ich vzájomné prepojenie. Prepojenie jednotlivých objektov je znázornené na obrázkoch 3.1. Prepojenie JukeboxController a aktivít prostredníctvom objektu Application je zobrazené na obrázku 3.2.

#### 3.1 Popis aplikačnej logiky

Obrázok 3.1 znázorňuje prepojenie jednotlivých tried v aplikačnej logike.



Obrázok 3.1: Triedny diagram zobrazujúci štruktúru aplikačnej logiky



---

### 3.1.1 JukeboxController

JukeboxController znázornený na obrázku 3.1 je objekt reprezentujúci aplikačnú logiku celého jukebox-u. Poskytuje rozhranie pre pridávanie multimédií do fronty a samotné prehrávanie. Registruje poslucháčov pre nasledujúce udalosti:

- Zmena aktuálne prehrávanej piesne (OnCurrentSongListener)
- Zmena fronty piesní (OnQueueChangeListener)
- Zmena pozície aktuálne hranej piesne (OnSongPositionListener)

Týchto poslucháčov implementuje objekt MainActivity a reaguje na ich volania aktualizáciou GUI. Nasledujúci zoznam demonštruje zoznam funkčností a objektov za dané funkčnosti zodpovedných:

- SongPlayer- Prehrávanie piesní, JukeboxController je registrovaný poslucháč pre dokončenie prehrávania, kde na túto udalosť reaguje spustením ďalšej piesne alebo spustením prehrávania ukážok v závislosti od stavu fronty piesní
- SamplePlayer- Prehrávanie hudobných ukážok, keď je fronta piesní prázdna
- CreditController- Manažuje stav kreditu
- MediaRetriever- Získanie multimediálneho obsahu (video, audio, obrázky albumov)
- VideoView- Slúži na prehrávanie videa, do JukeboxController je predaný z MainActivity
- TopSongsDataSource- Databáza najčastejšie prehrávaných piesní

Podrobný opis funkcií jednotlivých objektov je popísaný v nasledujúcich podkapitolách.

### 3.1.2 SongPlayer

SongPlayer zabezpečuje ovládanie prehrávania piesní a to spustiť respektíve zastaviť prehrávanie. Taktiež poskytuje informácie o aktuálnej pozícii prehrávanej piesne. Ďalej registruje poslucháčov pre dokončenie prehrávania piesne (OnCompletionListener).

### 3.1.3 SamplePlayer

Objekt SamplePlayer je zodpovedný za prehrávanie ukážkových piesní. V konštruktoře mu je predaná kolekcia všetkých piesní. Pokiaľ je fronta piesní prázdna JukeboxController spustí prehrávanie ukážok prostredníctvom SamplePlayer v náhodnom poradí. Pri pustení piesne je SamplePlayer zastavený.

### 3.1.4 AbstractCreditController

Je abstraktná trieda implementujúca základné operácie s kreditom ako napríklad inkrementáciu a dekrementáciu kreditu. Registruje poslucháčov pre zmenu kreditu. Trieda obsahuje dve abstraktné metódy connect() a disconnect() ktoré musí implementovať potomok. Takto navrhnutý model umožňuje viacero typov implementácie komunikácie s mincovníkom alebo iným zariadením, bez nutnosti väčšieho zásahu do kódu aplikácie.

### 3.1.5 CreditController

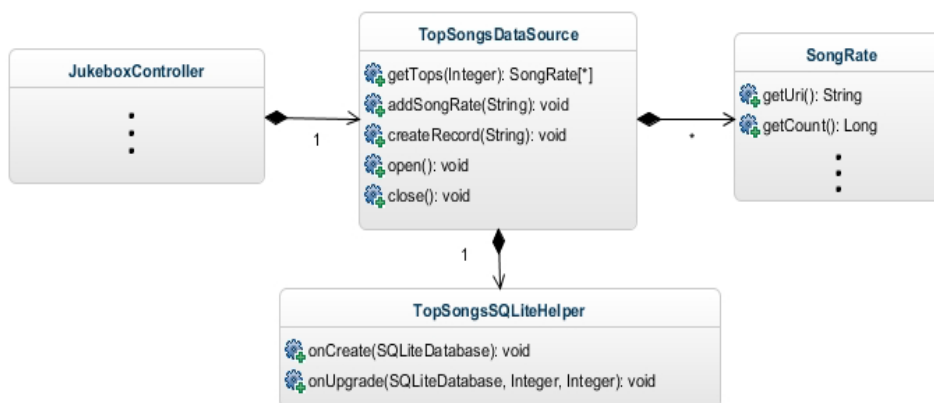
Je trieda rozširujúca AbstractCreditController. CreditController Implementuje sériovú komunikáciu s Anrduinom UNO. Pre túto komunikáciu využíva triedu SerialPort kde v samostatnom vlákne cyklicky overuje prichádzajúce dáta. Ak sa dáta zhodujú s prednastaveným reťazcom (Coin signal), pripočíta hodnotu Credit per pulse k aktuálnemu kreditu a oznámi poslucháčom zmeny kreditu túto udalosť.

### 3.1.6 MediaRetriever

Trieda MediaRetriever zodpovedá za získanie zoznamu dostupných multimediálnych súborov (video, audio a obrázky albumov). K tomu využíva vstavanú multimediálnu databázu systému Android, MediaStore. MediaRetriever vytvorí hierarchickú štruktúru Interpretov, Albumov a Piesní, ktorá je reprezentovaná kolekciou objektov Artist. Každý objekt typu Artist obsahuje kolekciu typu Album a každý objekt typu Album obsahuje kolekciu typu MediaEntity. Pre túto funkciu využíva triedu SongSorter.

### 3.1.7 TopSongDataSource

Zodpovedá za databázu najčastejšie prehrávaných piesní (TOP). Pre zjednodušenie triedneho diagramu nie je zobrazený na obrázku 3.1 ale osobitne na obrázku 3.2. Hodnotenie jednotlivých piesní je reprezentované objektom SongRate, ktorý nesie informáciu o URI piesne v String-ovej forme (primárny kľúč v tabuľke). Ďalšou informáciou ktorú objekt nesie je počet prehraní daného záznamu. Pokiaľ databáza neexistuje vytvorí sa pomocou objektu TopSongSQLiteHelper. Databáza obsahuje jednu tabuľku TABLE\_SONGS\_RATE, ktorá má dva stĺpce COLUMN\_URI a COLUMN\_COUNT. Záznamy v tabuľke sa aktualizujú pri každom pridaní piesne do fronty a to inkrementáciou počtu prehraní o 1. Ak záznam so zadaným URI neexistuje vytvorí sa s počiatočným počtom prehraní rovným 1. TopSongDataSource vracia kolekciu objektov SongRate, ktorých počet prehraní je najväčší. Žiadaný počet týchto hodnotení je predaný ako parameter.



Obrázok 3.2: Diagram tried znázorňujúci prepojenie JukeboxController a TopSongDataSource

### 3.1.8 Artist

Obsahuje kolekciu objektov typu Album, názov interpreta a id interpreta.

### 3.1.9 Album

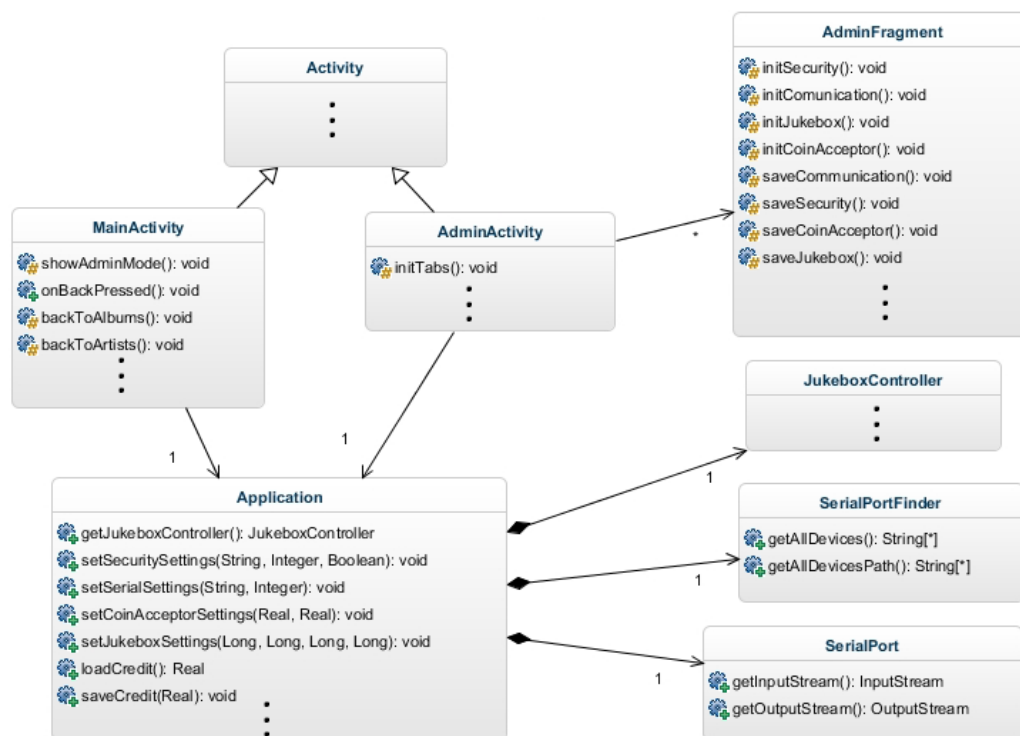
Obsahuje kolekciu objektov typu MediaEntity, názov albumu a id albumu.

### 3.1.10 MediaEntity

MediaEntity je trieda reprezentujúca multimedialný obsah. Tento obsah môže byť buď video alebo audio súbor. Objekt nesie informáciu o názve interpreta, názve albumu, názve titulu a URI. Ďalej MediaEntity možno dotazovať akého je typu. Na to slúži metóda isVideo(), ktorá vracia hodnotu typu Boolean.

## 3.2 Prepojenie aplikačnej logiky s aplikáciou Jukebox

Na nasledujúcom obrázku 3.3 je znázornené prepojenie JukeboxController s aktivitami prostredníctvom objektu Application.



Obrázok 3.3: Diagram tried zobrazujúci prepojenie aktivít s JukeboxController

---

### 3.2.1 MainActivity

MainActivity je aktivita, ktorá je zodpovedná za prepojenie aplikačnej logiky JukeboxController s GUI. Referenciu na JukeboxController získava prostredníctvom objektu Application.

### 3.2.2 AdminActivity

AdminActivity je aktivita, ktorá obsluhuje GUI administrátorského rozhrania. Ukladanie zmien prevedených v administratívnom rozhraní aktivita prevádza prostredníctvom referencie na objekt Application.

### 3.2.3 Application

Trieda Application rozširuje triedu android.app.Application. Jej funkcia okrem iného je nahratie nastavení pri spustení aplikácie a uloženie nastavení pri zmene. Okrem spomenutých nastavení Application ešte ukladá stav kreditu a stav fronty piesní pri ich zmene. Pri štarte sú tieto hodnoty načítané. Pre ukladanie informácií využíva SharedPreferences [4]. Taktiež vytvára inštancie objektov JukeboxController a CreditController a prístupňuje ich aktivitám.

## 3.3 Android SerialPort API

Keďže Android SDK nedisponuje žiadnym API pre prístup k linuxovým TTY sériovým portom bolo nutné použiť Android SerialPort API [8]. Tento projekt poskytuje jednoduché aplikačné rozhranie pre čítanie a zápis prostredníctvom sériového portu. Pre túto funkčnosť využíva JNI- Java Native Interface. Projekt je pod licenciou Apache License 2.0 [9].

Android SerialPort API je tvorené dvoma triedami: SerialPortFinder a SerialPort. Prepojenie týchto tried s aplikáciou je znázornené na obrázku 1.13.

### 3.3.1 Podporované funkcie API

- Zoznam dostupných sériových portov vrátane USB / serial adaptérov
- Konfigurácia týchto portov (baudrate, stop bit, práva)
- Poskytuje štandardný java InputStream a OutputStream

### 3.3.2 Nepodporované funkcie API

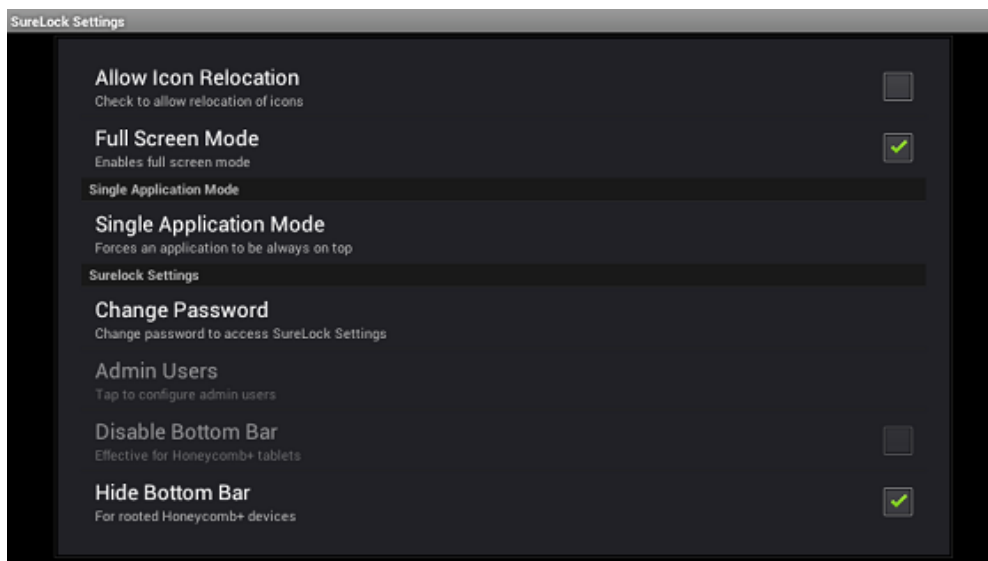
- Odosielanie a prijímanie dát prostredníctvom USB slave rozhrania

### 3.4 Vynútené spustenie a beh aplikácie Jukebox

O vynútené spustenie a skrytie System Bar-u sa stará aplikácia SureLock [5]. Túto aplikáciu vyvíja spoločnosť 42Gears Mobility Systems ktorá pôsobí v Indii meste Bangalore. Prvá verzia aplikácie bola vydaná v Decembri 2011. SureLock je primárne určený pre zabezpečenie zodpovedného používania zariadení, zvýšenie produktivity a zníženie nákladov na údržbu firemných zariadení. Tieto vlastnosti sú dosiahnuté povolením len žiadaných aplikácií a zamedzeniu užívateľovi meniť nastavenia systému. Aplikácia je naimplementovaná pre väčšinu platforiem (Windows 7/8, Windows Mobile, Android, iOS). Trial verzia pre Android je zdarma dostupná na Google Play, pričom niektoré funkčnosti sú deaktivované. Hlavnou nevýhodou trial verzie je, že neumožňuje zmenu hesla do administrátorského rozhrania. Plná licencia na jedno zariadenie v súčasnosti stojí okolo päťdesiat dolárov. Možnosti nastavení sú znázornené na obrázku 1.14.

#### 3.4.1 Funkčnosti aplikácie SureLock využité v Jukeboxe

- Single Application Mode- vynucuje beh jednej preddefinovanej aplikácie, ak aplikácia z nejakého dôvodu spadne, SureLock ju znovu spustí
- Full Screen Mode- vynucuje aplikácií beh v režime na celú obrazovku
- Hide Bottom Bar- skryje systémovú lištu a zamedzí tak užívateľovi k neoprávneným akciám



Obrázok 3.3: Ukážka nastavení aplikácie SureLock

---

## 4 Aplikácia Jukebox z užívateľského pohľadu

Aplikácia Jukebox má dve základné užívateľské rozhrania. Prvým je užívateľské rozhranie pre ovládanie prehrávania záznamov. Druhým je administrátorské rozhranie pre nastavovanie parametrov aplikácie. V nasledujúcich dvoch podkapitolách sú tieto rozhrania popísané.

### 4.1 Užívateľské rozhranie aplikácie Jukebox

V ľavej časti obrazovky sa nachádza zoznam, ktorý môže byť v troch úrovniach:

- Zobrazuje zoznam interpretov
- Zobrazuje zoznam albumov
- Zobrazuje zoznam piesní

Základná úroveň je zoznam interpretov. Po kliknutí na konkrétneho interpreta sa v zozname zobrazia albumy. Po kliknutí na konkrétny album zoznam zobrazí piesne prislúchajúce tomuto albumu. Pre navigáciu späť slúži tlačidlo Back umiestnené v ľavom dolnom rohu obrazovky.

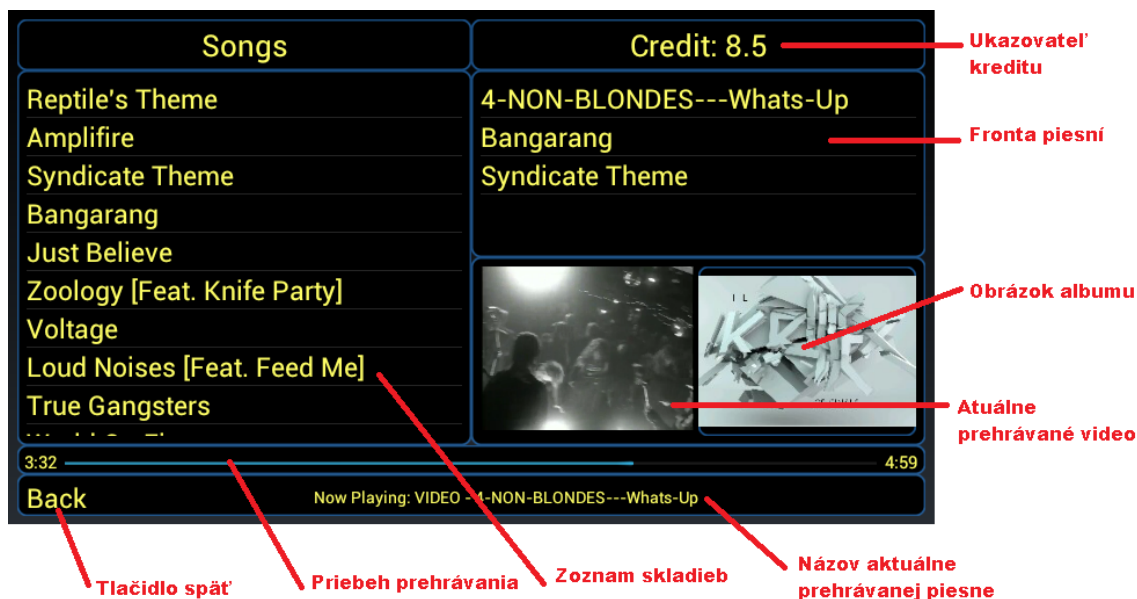
Pri kliknutí na pieseň sa zobrazí dialógové okno pre potvrdenie daného výberu. Tento dialóg sa zobrazí len pokiaľ je v jukebox-e dostatok kreditu pre prehranie piesne. V opačnom prípade sa zobrazí hlásenie o nedostatku kreditu. Pokiaľ užívateľ potvrdí výber, pieseň sa pridá do fronty respektíve ak je fronta prázdna, spustí sa prehrávanie. Za toto prehranie je odpočítaný kredit podľa nastavenej hodnoty Song cost.

Pri predĺženom kliku (LongClick) je pieseň pridaná do fronty prednostne to je na prvé miesto. Pokiaľ v jukebox-e nie je dostatok kreditu na prednostné prehranie nestane sa nič. Cena takéhoto uprednostnenia je dvojnásobok ceny normálneho prehrávania. Pokiaľ je fronta prázdna, odpočíta sa cena za normálne prehranie.

V pravej časti obrazovky sa nachádza fronta piesní, ktoré budú postupne prehrávané. Pod frotou piesní sú umiestnené dve polia. Ľavé pole slúži na zobrazenie prehrávaného videa a pravé pole zobrazuje obrázok prislúchajúci aktuálnemu albumu. Obrázok albumu sa zobrazuje iba pokiaľ sa užívateľ nachádza v zozname piesní daného albumu. Nasledujúci obrázok 1.15 znázorňuje toto rozloženie.

V spodnej časti obrazovky sú umiestnené dve nad sebou uložené lišty. Prvá vyššie uložená lišta indikuje priebeh práve prehrávaného záznamu. Druhá nižšie uložená lišta obsahuje na ľavej strane tlačidlo Back a v strede tejto lišty je umiestnený názov interpreta spolu s názvom titulu práve prehrávanej piesne.

Toto rozloženie je graficky znázornené na obrázku 4.1 na nasledujúcej strane.



Obrázok 4.1: Ukážka užívateľského rozhrania aplikácie Jukebox

## 4.2 Administrátorské rozhranie aplikácie Jukebox

Pre vstup do administrátorského rozhrania je potrebné opakovane klikat' na Label Credit. Počet týchto klikov je možné nadefinovať v administrátorskom menu, záložka SECURITY. Po dostatočnom počte opakovaní sa zobrazí dialóg, ktorý užívateľa žiada o vloženie hesla. Pokiaľ užívateľ zadá toto heslo správne, aplikácia mu zobrazí administrátorské menu. Pri zadaní nesprávneho hesla sa dialóg skryje.

Administrátorské rozhranie je rozdelené do štyroch záložiek:

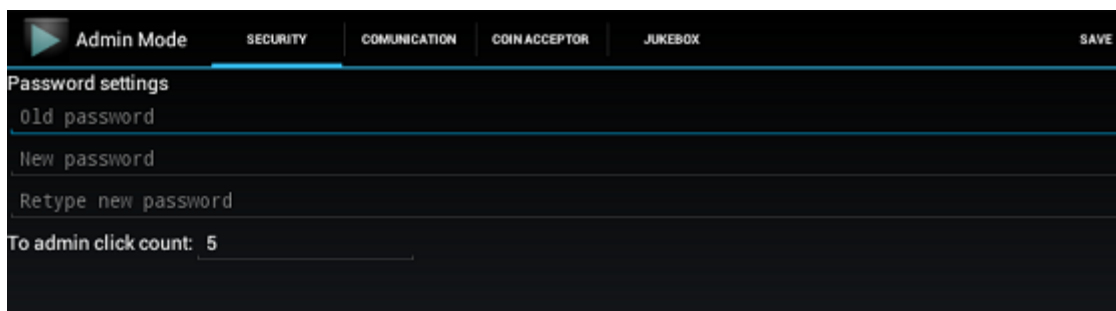
- SECURITY
- COMUNICATION
- COIN ACCEPTOR
- JUKEBOX

Na každej záložke je možné nastavovať špecifické hodnoty. Tieto hodnoty sú popísané v nasledujúcich podkapitolách. Pre uloženie zmenených hodnôt v administrátorskom menu užívateľ klikne na tlačidlo SAVE v pravom hornom rohu obrazovky.

Pre navigáciu späť do užívateľského režimu slúži tlačidlo v ľavom hornom rohu obrazovky s ikonou aplikácie.

### 4.2.1 Záložka SECURITY

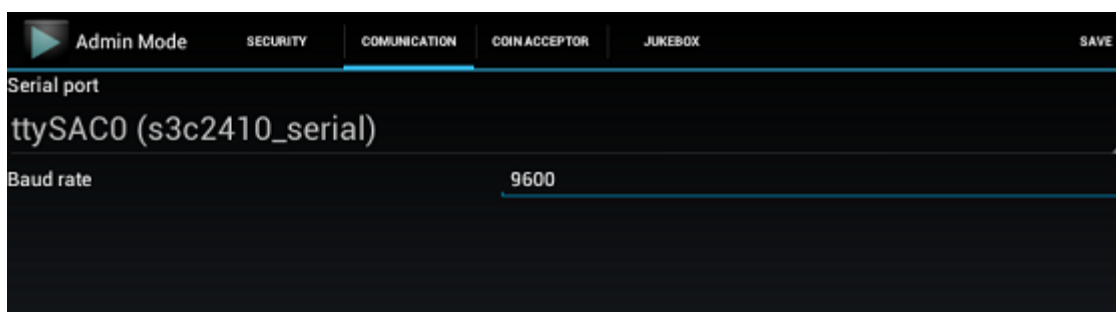
V záložke SECURITY (obrázok 4.2) má užívateľ možnosť zmeniť prístupové heslo pre prístup do administrátorského rozhrania. Je nutné zadať stávajúce heslo. Ďalej užívateľ môže nadefinovať počet kliknutí na Label Credit, po ktorom sa zobrazí dialóg na zadanie hesla do administrátorského rozhrania.



Obrázok 4.2: Záložka SECURITY

#### 4.2.2 Záložka COMMUNICATION

V záložke COMMUNICATION (Obrázok 4.3) je možné vybrať prostredníctvom ktorého sériového portu má aplikácia komunikovať s Arduino. Ďalej poskytuje možnosť zmeny baud rate, čiže rýchlosti prenosu.

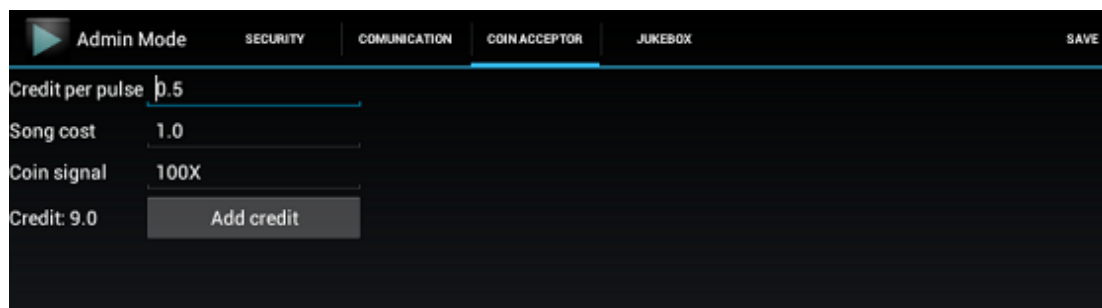


Obrázok 4.3: Záložka COMMUNICATION

#### 4.2.3 Záložka COIN ACCEPTOR

Na tejto záložke (obrázok 4.4) je možné nastaviť hodnoty:

- Credit per pulse- o akú čiastku má byť navýšený kredit pri jednom pulze
- Song cost- cena piesne
- Coin signal- reťazec, na ktorý bude aplikácia reagovať na sériovom porte ako na podnet pre navýšenie kreditu
- Tlačidlo Add credit- pri jeho stlačení sa kredit navýši o hodnotu Credit per pulse



Obrázok 4.4: Záložka COIN ACCEPTOR

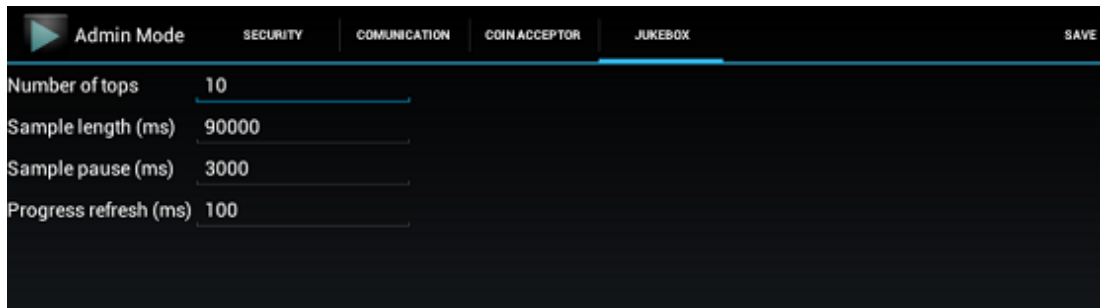


---

#### 4.2.4 Záložka JUKEBOX

Na záložke JUKEBOX (obrázok 4.5) je možné nastavovať hodnoty:

- Number of tops- počet piesní v TOP liste
- Sample length(ms)- dĺžka trvania jednej ukážky v milisekundách
- Sample pause(ms)- dĺžka trvania pauzy medzi ukážkami v milisekundách
- Progress refresh(ms)- hovorí, v akých intervaloch sa majú obnoviť ukazovatele stavu prehrávania v milisekundách



Parameter	Value
Number of tops	10
Sample length (ms)	90000
Sample pause (ms)	3000
Progress refresh (ms)	100

Obrázok 4.5: Záložka JUKEBOX

Ukázkové obrázky boli upravené tak, aby zobrazovali len podstatnú časť obrazovky.

---

## 5 SOZA- Slovenský ochranný zväz autorský

Slovenský Ochranný Zväz Autorský (ďalej len SOZA) [6] je právnická osoba, neziskové občianske združenie autorov a vydavateľov hudobných diel. Na území Slovenska spravuje majetkové autorské práva k hudbe z celého sveta, vykonáva kolektívnu správu práv podľa Autorského zákona. SOZA má uzavretých 63 medzinárodných recipročných zmlúv o vzájomnom zastúpení, na základe ktorých zabezpečuje uplatňovanie práv slovenských hudobných tvorcov vo všetkých krajinách sveta, v ktorých existuje autorsko-právna ochrana. Týmto bilaterálnymi zmluvami s partnerskými ochrannými organizáciami má pokrytý prakticky celý svet. Recipročne SOZA zabezpečuje na území SR uplatňovanie práv zahraničných tvorcov rovnocenne právam domácich autorov. SOZA je uznávaným subjektom aj na medzinárodnej scéne. Je plnoprávnym členom dvoch veľkých celosvetových organizácií – CISAC a BIEM.

### 5.1 Sadzobník pre hudobné automaty platný v roku 2014

Nasledujúce dve tabuľky zobrazujú ceny mesačných poplatkov za jednotlivé kategórie prevádzkovania hudobných automatov:

Kategória	Paušál v € bez DPH / časová jednotka
1 - 50 kusov	16,60 € / mesiac
51 - 100 kusov	15,00 € / mesiac
Nad 100 kusov	13,30 € / mesiac

Tabuľka 5.1: Hudba z CD hudobných automatov len na počúvanie

Kategória	Paušál v € bez DPH / časová jednotka
1 - 50 kusov	19,90 € / mesiac
51 - 100 kusov	18,30 € / mesiac
Nad 100 kusov	16,60 € / mesiac

Tabuľka 5.2: Hudba z digitálneho zdroja (MP3) hudobných automatov len na počúvanie

---

## 5.2 Špeciálne označenie hudobných automatov

Každý hudobný automat umiestnený vo verejnosti prístupnom priestore musí byť označený licenčnou známkou SOZA s nalepeným hologramom a registračným číslom. Hudobný automat bez licenčnej známky je nelegálny.

Prevádzkovateľ je povinný požiadať SOZA o vydanie licenčnej známky na každý nový hudobný automat najmenej 10 dní pred začatím ich verejnej prevádzky a to k 1. alebo k 15. dňu v mesiaci. Uvedená lehota je nevyhnutná na spracovanie a doručenie licenčných známok prevádzkovateľovi. Licenčné známky vydá SOZA bezplatne. Formulár Žiadosti o vydanie licencie tvorí prílohu č.2 k Hromadnej licenčnej zmluve.

Licenčné známky je prevádzkovateľ povinný umiestniť na viditeľné miesto (napr. vpredu pod krycie sklo alebo plexisklo pri obrazovke alebo u niektorých typov nalepiť zvonku alebo na zadnú stenu a ochrániť priskrutkovaným plexisklom) a zabezpečiť ich pred poškodením alebo odlepením. Majiteľ, resp. prevádzkovateľ hudobného automatu je povinný oboznámiť osoby zodpovedné za prevádzku s umiestnením licenčnej známky.

Ak inšpektor SOZA zistí, že hudobný automat je bez licenčnej známky, majiteľ, resp. prevádzkovateľ zaplatí základnú sadzbu autorskej odmeny za všetky neregistrované zariadenia bez licenčnej známky, ktoré odhalia inšpektori alebo orgány činné v trestnom konaní.

Výroba a prevádzkovanie hudobného automatu bez licencie SOZA a bez zaplatenie autorských odmien je nedovoleným nakladaním s dielami chránenými Autorským zákonom č.618/2003 Z.z. Porušovanie autorských práv je sankcionované podľa § 283 Trestného zákona.

---

## Záver

Cieľom tejto práce bolo vytvoriť Android jukebox, ktorý komunikuje s elektronickým mincovníkom ako po hardvérovej, tak aj po softvérovej stránke. Zadanie práce som splnil a vytvoril zariadenie na platforme Android, ktoré je schopné prehrávať audio a aj video záznamy a zároveň reaguje na vhodenie mince pripočítaním kreditu.

Pri písaní aplikácie Jukebox som sa stretol s nasledujúcim problémom. Android SDK neobsahuje aplikačné rozhranie pre sériovú komunikáciu. Problém som vyriešil pridaním Android SerialPort API do aplikácie. Toto aplikačné rozhranie umožňuje komunikáciu prostredníctvom sériového portu ako aj vyhľadávanie dostupných portov na zariadení. Danú funkčnosť vykonáva prostredníctvom JNI.

Aplikácia Jukebox bola testovaná na zariadeniach: FriendlyARM Smart210 (Android 4.0.3), Gsmart Roma R2 (Android 4.2.2) a Nexus 7 (Android 4.4.2). Pričom na posledných dvoch zmienených zariadeniach nebola testovaná komunikácia s mincovníkom z dôvodu neprítomnosti sériového portu.

Ďalšie možné rozšírenia tohto zariadenia sú napríklad prehrávanie streamovaného obsahu z internetu, vytvorenie klientskej mobilnej aplikácie pre vzdialené ovládanie jukebox-u alebo implementácia viacerých možností pripojení mincovníka, ako napríklad prostredníctvom USB host alebo Bluetooth.

---

## Použitá literatúra

- [1] FRIENDLYARM. Smart210 [online]. [cit. 2014-05-01]. Dostupné z: <http://www.friendlyarm.net/products/smart210>
- [2] ARDUINO. Arduino UNO [online]. [cit. 2014-05-01]. Dostupné z: <http://arduino.cc/en/Main/ArduinoBoardUno>
- [3] CRANE PAYMENT INNOVATIONS. CcTalk [online]. 1996 [cit. 2014-04-01]. Dostupné z: <http://www.cctalk.org/>
- [4] ANDROID DEVELOPERS [online]. [cit. 2014-05-01]. Dostupné z: <http://developer.android.com/>
- [5] 42GEARS MOBILITY SYSTEMS. SureLock [online]. [cit. 2014-05-01]. Dostupné z: <http://www.42gears.com/surelock/>
- [6] SOZA. Slovenský ochranný zväz autorský [online]. [cit. 2014-04-28]. Dostupné z: <http://www.soza.sk/>
- [7] IBM. UML basics: The class diagram [online]. [cit. 2014-05-04]. Dostupné z: <https://www.ibm.com/developerworks/>
- [8] GOOGLE DEVELOPERS. Android-serialport-api [online]. [cit. 2014-05-04]. Dostupné z: <https://code.google.com/p/android-serialport-api/>
- [9] THE APACHE SOFTWARE FOUNDATION. Apache License, Version 2.0 [online]. 2004 [cit. 2014-05-04]. Dostupné z: <http://www.apache.org/licenses/LICENSE-2.html>

---

## **Zoznam príloh**

1. Text bakalárskej práce vo formáte PDF: bakalarska\_praca.pdf
2. Vyexportovaný projekt z Eclipse v podobe zip: Jukebox.zip
3. Vytvorený .apk súbor aplikácie Jukebox: Jukebox.apk